

# ETL & Data Ware House

## Group member

Evan(Seung-Gyeom) KIM  
Yingpeng YANG  
Hongbo YE

DA ACC A03  
DA ALT A03  
DA ALT A03



**School of  
Engineering**

# TABLE OF CONTENTS

I.	INTRODUCTION AND DATA	3
	Call Data	3
	Call Types	3
	US States	3
	Employees	3
	Call Charges	3
	3 Steps ETL & Data Ware Housing	3
II.	STAGING DATABASE	4
	II-1.Call Data Table	4
	II-2. Call Types Table	8
	II-3. US State Table	8
	II-4. Employees Table	10
	II-5. Call Charges Table	11
III.	OPERATIONAL DATA STORE	12
	III-1. Call Charge Table	12
	III-2. Call Data Table	15
	III-3. Employees Table	22
IV.	DATA WARE HOUSE	25
	IV-1. Date Dimension Table	25
	IV-2. Employees Dimension Table	25
	IV-3. Call Data Fact Table	28
V.	Use case	33
VI.	CONCLUSION	38

# I. INTRODUCTION AND DATA

ServiceSpot, an IT company, needs an assistance to analyze their call center data. They provided their daily calls data hoping to improve their quality of service. The mission is to develop an EPL project with SSIS to conduct the data warehouse.

## Call Data

Constitute 3 cvs files for 3 years of data 2018-2020 having same 6 columns

- **CallTimestamp** : Date and Time
- **Call Type** : Type of calls
- **EmployeeID** : Foreign key form Employees table
- **CallDuration** : Counts by second
- **WaitTime** : Counts by second
- **CallAbandoned** : 0 and 1 Boolean value

## Call Types

- **CallTypeID** : 1-3
- **CallTypeLabel** : label of call types

## US States

- **StateCD** : 2-letter state code
- **Name**: Name of the state
- **Region**: East, West, etc

## Employees

- **EmployeeID** : Employ unique identifier
- **EmployeeName** : Full name of employee
- **Site** : Location of office
- **ManagerName** : Employee's supervisor

## Call Charges

- **CallType** : label of call types
- **Call Charge** : charge per minute, 4 columns for each 4 years of charge evolution 2018-2021

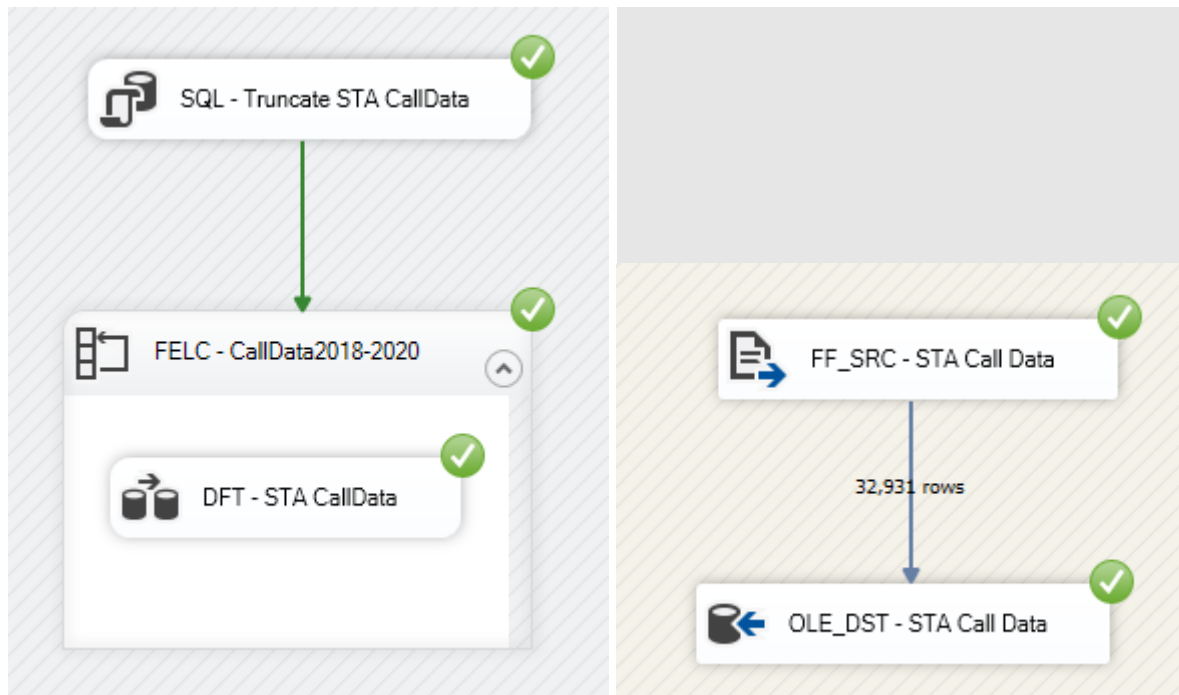
## 3 Steps ETL & Data Ware Housing

- **STA, Staging phase** :  
Load and explore all the data
- **ODS, Operational data store phase** :  
Clean and standardize the data.  
Create a "Technical\_Rejects" table in case of issue.
- **Data Ware House Phase** :  
Organize one fact table. Create a "Functional\_Reject" table in case of issue.

## II. STAGING DATABASE

At the data staging phase, the length is globally set to 255 as a precaution. It prevents errors when the data has an excessive length. The Code page is globally set to “1252 (ANSI - Latin I)” Since the “Data Flow Task”, the “OLE DB Destination” and the “Execute SQL Task” are iterative, they will be described only once.

### II-1.Call Data Table



Results Messages

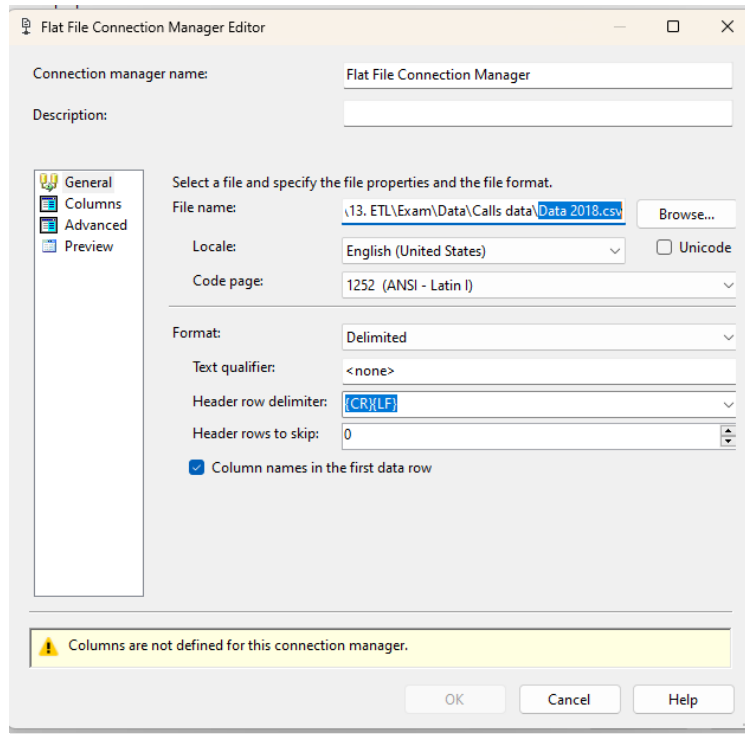
	CallTimestamp	Call Type	EmployeeID	CallDuration	WaitTime	CallAbandoned
1	1/1/2018 10:06	2	S704443	1189	0	0
2	1/1/2018 10:21	2	U641256	416	18	0
3	1/1/2018 10:22	2	T398672	1473	0	0
4	1/1/2018 10:31	2	X825147	579	0	0
5	1/1/2018 10:31	1	M855788	886	17	0

Query executed successfully. | MSI (15.0 RTM) | AzureAD\EvanKim (56) | Report\_CallCenter\_STA | 00:00:01 | 98,975 rows

- Remark :** In order to count the number of rows, “Get Data” function was used in Excel : (Data 2018.csv, 33058 rows | Data 2019.csv, 32988 rows | Data 2020.csv, 32932 rows) The sum of rows counts is 98978. The difference between them is 3 (98978 - 98975 = 3). Assuming in SSIS, each table’s header row isn’t counted therefore, the result is correct.
- Solution for 3 CVS files :** The Call Data covers 3 years of data, during 2018-2020. Each file contains one year of Call Data, making a total of 3 files. “Foreach Loop Container” will be used to gather 3 different tables into one table “CallData”. First, proceed with

data staging as usual case, then add “Foreach Loop Container” in “Control Flow” to gather those files.

- **DFT - STA Call Data** : Drag and drop “Data Flow Task” SSIS Tool on “Control Flow” board.



**FF\_SRC - STA Call Data** :

Drag and drop “Flat File Source” component into “Data Flow” →

Click “Browse” to select the file “Data 2018.csv” → Code page : 1252 (ANSI - Latin I), allows to load data without error and gives data type “varchar” → go to Advanced menu then set the OutputColumnWidth to 255

- **OLE\_DST - STA Call Data** : Drag and drop “OLE DB Destination” component in “Data Flow” → Connect to SQL Server → Create “CallData” table.

```
CREATE TABLE [CallData] (
    [CallTimestamp] varchar(255),
    [Call Type] varchar(255),
    [EmployeeID] varchar(255),
    [CallDuration] varchar(255),
    [WaitTime] varchar(255),
    [CallAbandoned] varchar(255)
)
```

- **FELC- CallData2018-2020** : Add a new variable “FilePath” and set the Data type “String”. As value, paste the path of the file “Data 2018.csv”. It sets the file path to search right file path when “Foreach Loop Container” will be executed.

Variables					
Name	Scope	Data type	Value	Expression	
FilePath	STA_CallData	String	lata\Data 2018.csv		...

Drag and Drop “Foreach Loop Container” component → Collection → Select “Foreach File Enumerator” → Set the path of Folder → set the iteration reference as every csv format file in the Folder

The Foreach Loop container allows execution iteration over an enumeration.

**Foreach Loop Editor**

Enumerator: **Foreach File Enumerator**

Expressions:

**Enumerator**  
Specifies the enumerator type.

**Enumerator configuration**

Folder: C:\Users\EvanKim\iCloudDrive\DSTI\13. ETL\Exam\Data\Calls data

Files: \*.csv

Retrieve file name  
☐ Name and extension ☒ Fully qualified ☐ Name only

☐ Traverse subfolders

OK Cancel Help

Variable Mappings → Select “User::FilePath”

General

Collection

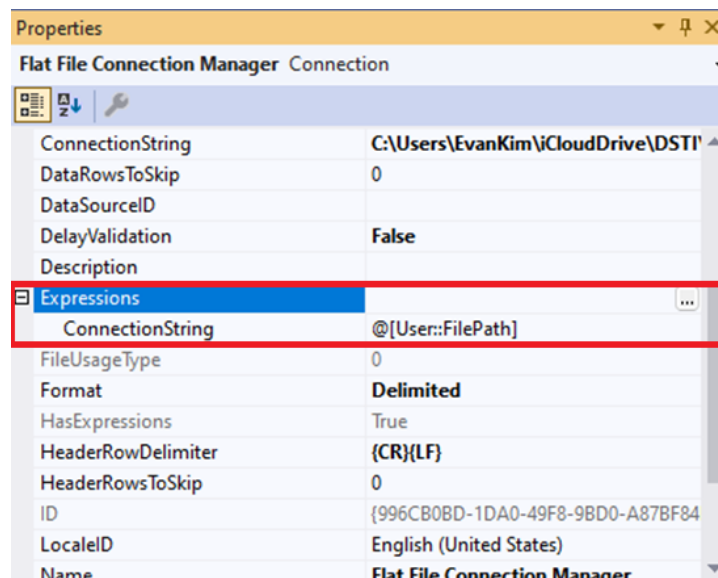
Variable Mappings

Expressions

Select variables to map to the collection value.

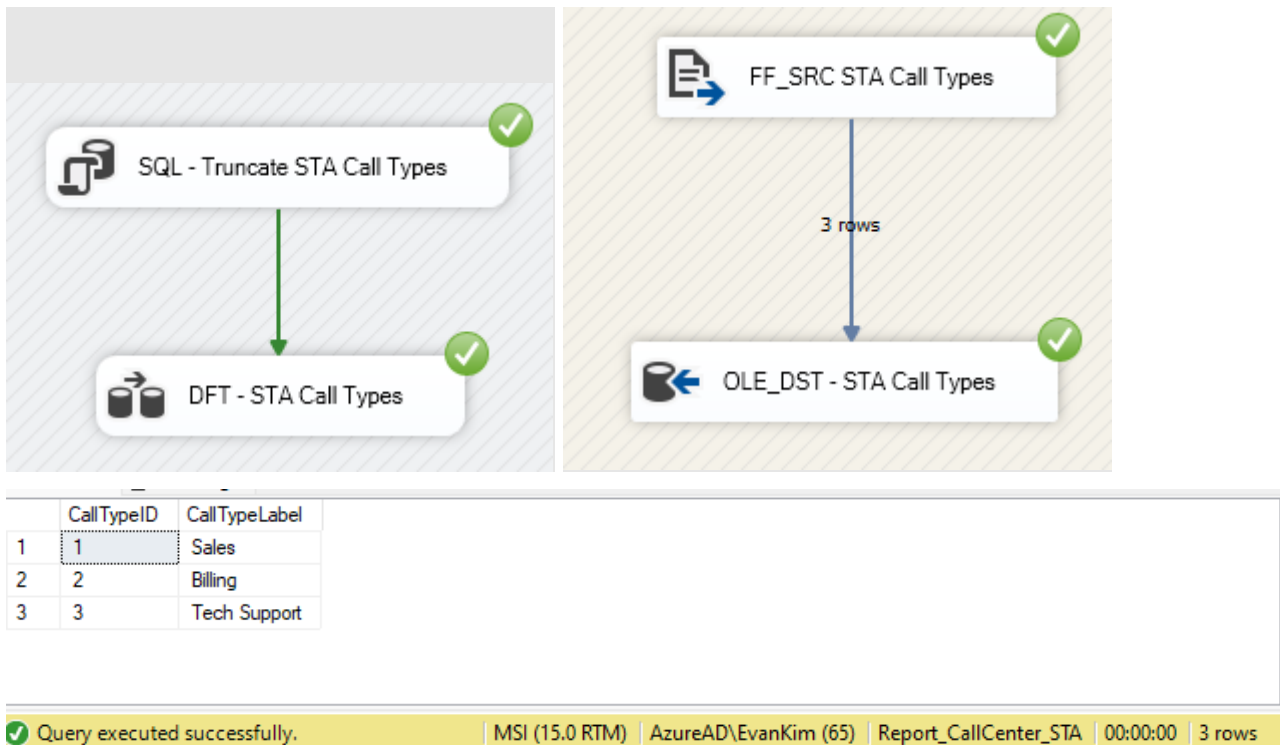
Variable	Index
User::FilePath	0

Go to Properties setting for Flat File Connection Manager → Add an expression selecting “ConnectionString” then “[User::FilePath]” as variable



- **SQL - Truncate STA Call Data** : Drag and drop “Execute SQL Task” SSIS component in “Control Flow”. Type the SQL Statement “TRUNCATE TABLE dbo.CallData”.

## II-2. Call Types Table



- OLE\_DST - STA Call Types :

```

1 CREATE TABLE [Call Types] (
2     [CallTypeID] varchar(255),
3     [CallTypeLabel] varchar(255)
4 )
  
```

## II-3. US State Table





	StateCD	Name	Region
1	AK	Alaska	West
2	AL	Alabama	South
3	AR	Arkansas	South
4	AZ	Arizona	West
5	CA	California	West
6	CO	Colorado	West
7	CT	Connecticut	Northeast
8	DC	District of Columbia	South
9	DE	Delaware	South
10	FL	Florida	South
11	GA	Georgia	South
12	HI	Hawaii	West
13	IA	Iowa	Midwest
14	ID	Idaho	West
15	IL	Illinois	Midwest
16	IN	Indiana	Midwest
17	KS	Kansas	Midwest
18	KY	Kentucky	South
19	LA	Louisiana	South

Query executed successfully. | MSI (15.0 RTM) | AzureAD\EvanKim (66) | Report\_CallCenter\_STA | 00:00:00 | 51 rows

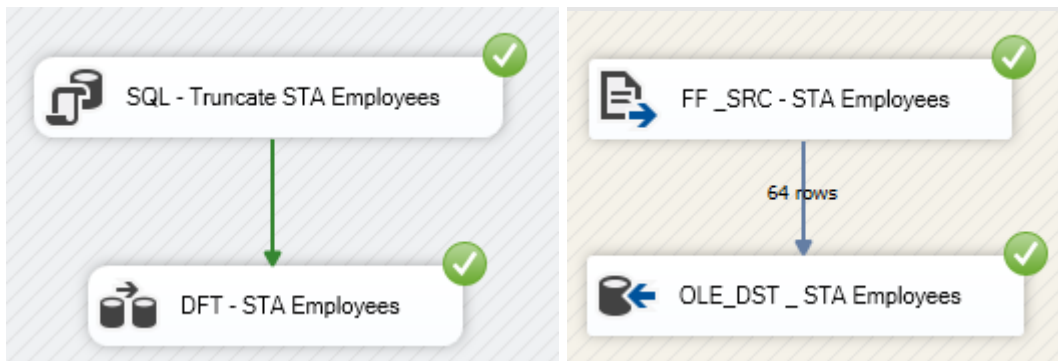
- OLE\_DST - STA US States :

```

1 CREATE TABLE [USstates] (
2     [StateCD] varchar(255),
3     [Name] varchar(255),
4     [Region] varchar(255)
5 )

```

## II-4. Employees Table



	EmployeeID	EmployeeName	Site	ManagerName
1	N772493	Onita Trojan	Spokane, WA	Deidre Robbs
2	F533051	Stomy Seller	Aurora, CO	Elsie Taplin
3	S564705	Mable Ayoub	Aurora, CO	Shala Lion
4	I281837	Latrisha Buckalew	Aurora, CO	Rana Taub
5	Y193775	Adrianna Duque	Spokane, WA	Collin Trotman
6	J632516	Keiko Daulton	Spokane, WA	Jamar Pahl
7	G727038	Dolores Lundeen	Aurora, CO	Shala Lion
8	V126561	Wilbur Mohl	Jacksonville, FL	Casey Bainbridge
9	E243130	Ileen Bomstein	Jacksonville, FL	Gonzalo Lesage
10	C206355	Janeth Roesler	Spokane, WA	Miyoko Degraw
11	G586239	Shery Hover	Aurora, CO	Elsie Taplin
12	M163408	Trevor Cerda	Spokane, WA	Collin Trotman
13	H438047	Debora Wilker	Spokane, WA	Collin Trotman
14	T659609	Kelvin Nicoletti	Spokane, WA	Jamar Pahl
15	P286634	Rafael Langton	Aurora, CO	Elsie Taplin
16	T387561	Alla Winkel	Jacksonville, FL	Ardath Duchame
17	J419954	Harrison Finlayson	Aurora, CO	Rana Taub
18	N470942	Caterina Jantz	Spokane, WA	Miyoko Degraw
19	B971624	Agripina Snively	Aurora, CO	Elsie Taplin

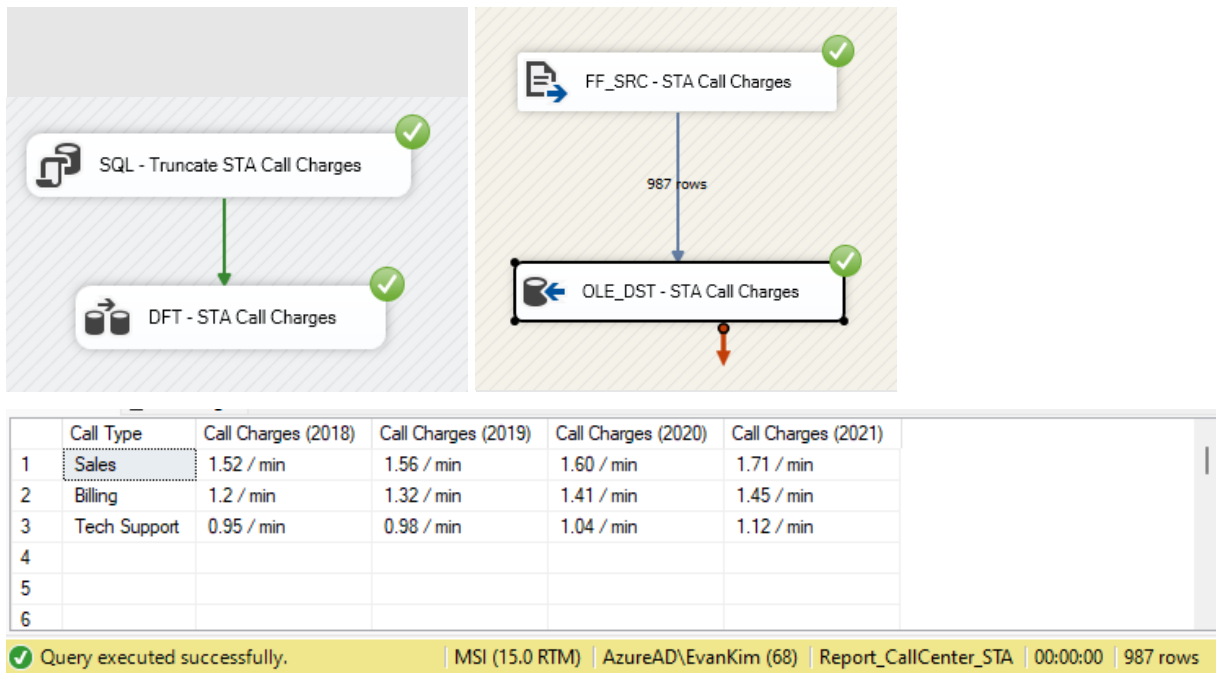
Query executed successfully. | MSI (15.0 RTM) | AzureAD\EvanKim (63) | Report\_CallCenter\_STA | 00:00:00 | 64 rows

- OLE\_DST - STA Employees :

```

1 CREATE TABLE [Employees] (
2     [EmployeeID] varchar(255),
3     [EmployeeName] varchar(255),
4     [Site] varchar(255),
5     [ManagerName] varchar(255)
6 )
  
```

## II-5. Call Charges Table



- **Remark :** The original Call Charges.csv file has 987 rows on the other hand, the essential data is only 3 rows. Despite of the empty rows, there was no impact on the result because the Lookup SSIS component doesn't recognize the empty data. It matches only the rows where it can refer to the joinable keys.

- OLE\_DST - STA Call Charges :

```

1 CREATE TABLE [CallCharges] (
2     [Call Type ] varchar(255),
3     [Call Charges (2018)] varchar(255),
4     [Call Charges (2019)] varchar(255),
5     [Call Charges (2020)] varchar(255),
6     [Call Charges (2021)] varchar(255)
7 )
8

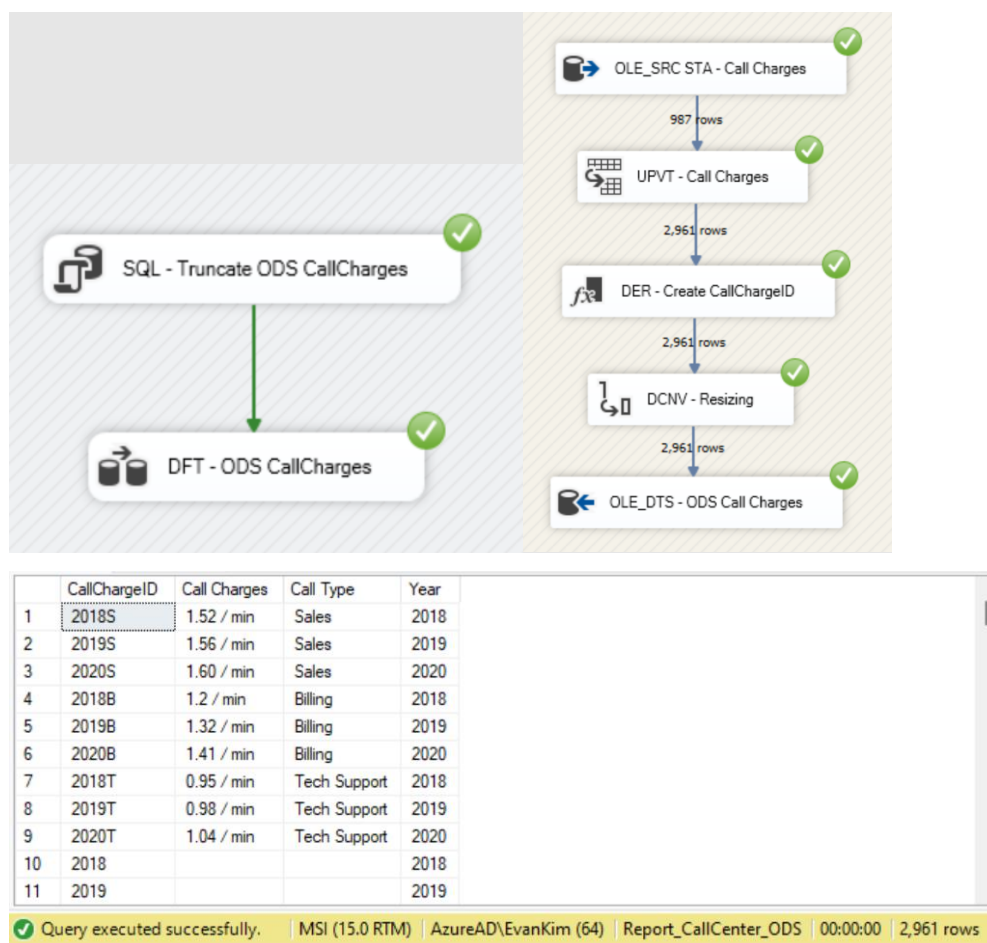
```

### III. OPERATIONAL DATA STORE

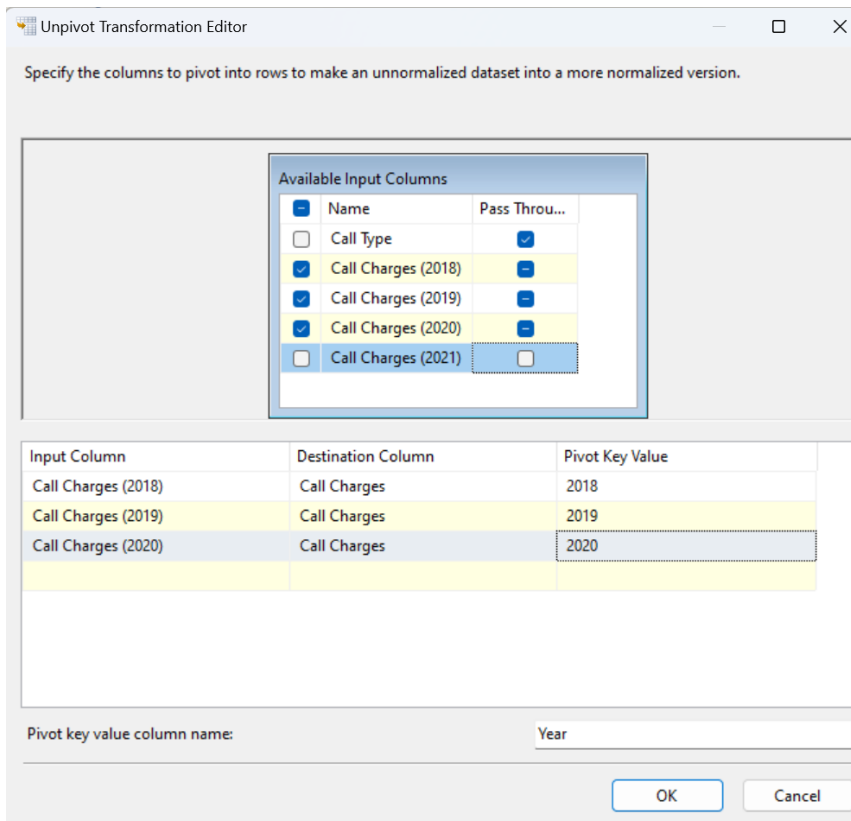
Taking into account the anticipation for the 2021 data, the current focus revolves around the framework for processing new data through ETL (Extract, Transform, Load) process. The “Execute SQL Task” and “OLE DB Source” tanks are iterative, they will be described only once.

#### III-1. Call Charge Table

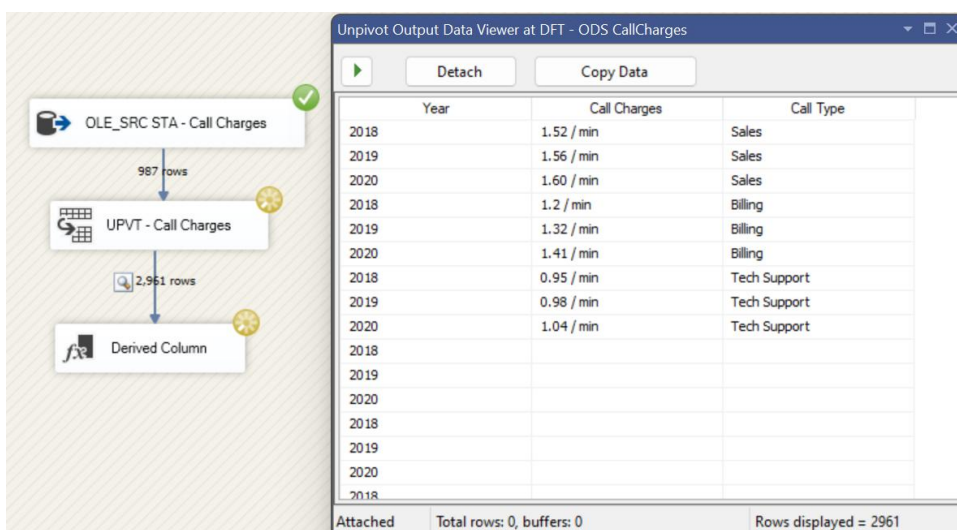
Using the “Unpivot” to transform Call Charges table into operational form for ETL process.



- **OLE\_SRC STA - Call Charges** : Drag and drop “OLE DB Source” SSIS component → load Call Charges data from SQL Data base where the data of Call Chares is stored.
- **UPVT - Call Charges** : Drag and drop “Unpivot” SSIS component → Select the “Available Input Columns” → Destination column : “Call Charges” → Pivot Key Value : 2018, 2019, 2020 → Pivot key value column name : “Year”



- Select the input column except 2021 (the 2021 data is not yet validated)
- If the Pass Through check box is marked, it will not be unpivoted but will instead be passed through to the output without any transformation.
- Destination Column refers to the column in the output data flow where the values from the unpivoted columns will be placed.
- Pivot Key Value refers to unique identifies for each rows in the unpivoted data. It differentiate between rows and determine which values belong to which original.
- Data Viewer



- **DER - Create CallChargeID** : Drag and drop “Derived Column” SSIS component →  
Derived Column Name : CallChargeID , Expression : (DT\_STR,4,1252)LEFT(Year,4) + (DT\_STR,4,1252)LEFT([Call Type],1)

Derived Column Name	Derived Column	Expression	Data Type	Length	Precision	Scale	Code Page
CallChargeID	<add as new column>	(DT_STR,25,1252)LEFT(Year,4) + (DT_STR,25,1252)LEFT([Call Type],1)	Unicode string [DT_WSTR]	50			

This table doesn't have any joinable key, in this case, it's necessary to create a new kind of primary key to join the fact table.

The diagram shows a Data Flow Task with the following components and row counts:

- OLE\_SRC STA - Call Charges**: 987 rows
- UPVT - Call Charges**: 2,961 rows
- DER - Create CallChargeID**: 2,961 rows
- Data Conversion**: 2,961 rows

The **Derived Column Output Data Viewer at DFT - ODS CallCharges** shows the following data:

Year	Call Charges	Call Type	CallChargeID
2018	1.52 / min	Sales	2018S
2019	1.56 / min	Sales	2019S
2020	1.60 / min	Sales	2020S
2018	1.2 / min	Billing	2018B
2019	1.32 / min	Billing	2019B
2020	1.41 / min	Billing	2020B
2018	0.95 / min	Tech Support	2018T
2019	0.98 / min	Tech Support	2019T
2020	1.04 / min	Tech Support	2020T
2018			2018

Attached | Total rows: 0, buffers: 0 | Rows displayed = 2961

- **DCNV - Resizing** : Drag and Drop “Data Conversion” SSIS component → Select the columns to resize the length

Input Column	Output Alias	Data Type	Length	Precision	Scale	Code Page
Call Charges	Call Charges_R	string [DT_STR]	15			1252 (ANSI - Latin I)
Call Type	Call Type_R	string [DT_STR]	13			1252 (ANSI - Latin I)
Year	Year_R	Unicode string [DT_WSTR]	7			

- **OLE\_DTS - ODS Call Charges** :

```

1 CREATE TABLE [CallCharges] (
2   [CallChargeID] nvarchar(8),
3   [Call Charges] varchar(15),
4   [Call Type] varchar(13),
5   [Year] nvarchar(7)
6 )

```

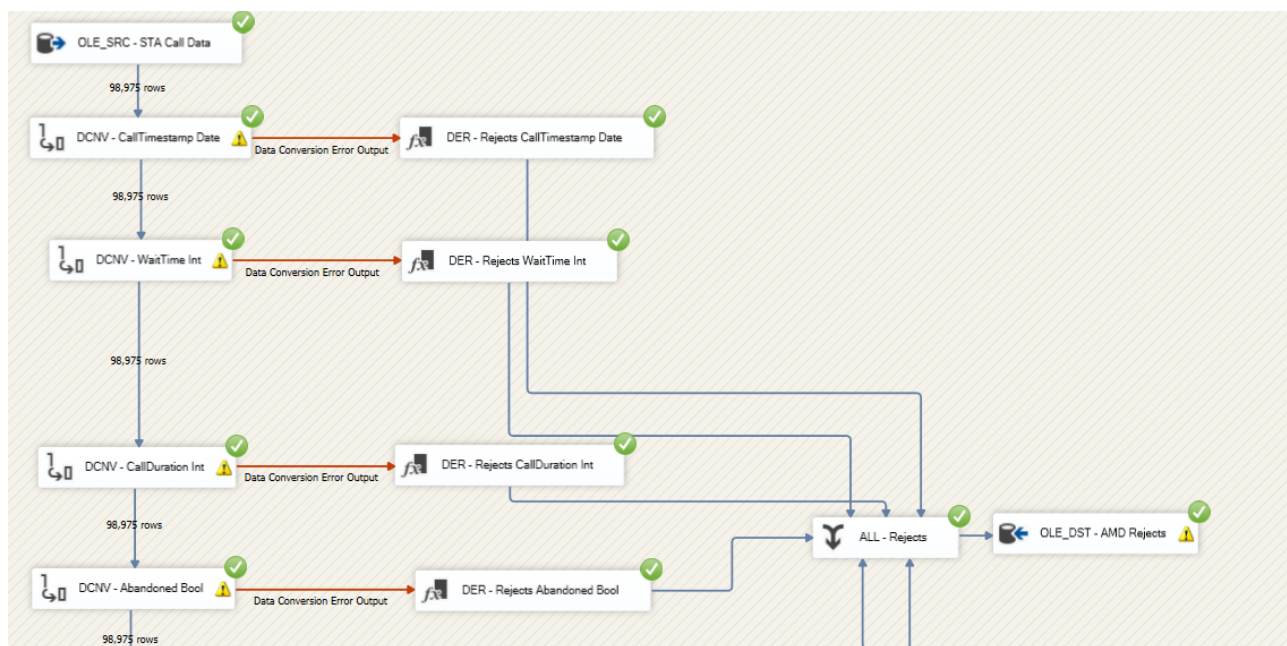
## III-2. Call Data Table

Pursuit to the STA phase :

1. "CallDuration" and "WaitTime" : will be cast into integer data type.
2. CallTimestamp : has a string data type and it's similar to the datetime format such as "mm-dd-yyyy hh:mm". Several data conversions will be anticipated. For instance, cast into datetime data type and separate into "yyyy-dd" and "hh:mm" creating two new different columns.
3. CallTypes : has only one byte of integer value. The Lookup SSIS component will allow obtaining the label of Call Types from "CallTypes" table. The data convention is not required, there is no issue joining with CallData type.
4. SLA Status : the standard call waiting time is the 35 scends. If the call was answered within 35 seconds, "Within SLA" will appear otherwise "Outside SLA" will appear.
5. EmployeeID : an adjustment of length will be applied.
6. CallAbandoned : will be cast into the Boolean data type.

### Part 1 : Data Conversion and Technical Rejects

Before joining CallTypes table with CallData table, cast the data type for "CallTimestamp", "CallDuration" and "CallAbandoned" columns. At the end of this process, the redirected error columns will appear in the table named "Technical\_Rejects" on the ADM SQL database server.





RejectDate RejectPackageAndTask RejectColumn RejectDescription

Query executed successfully. MSI (15.0 RTM) AzureAD\EvanKim (66) Report\_CallCenter\_ADM 00:00:00 0 rows

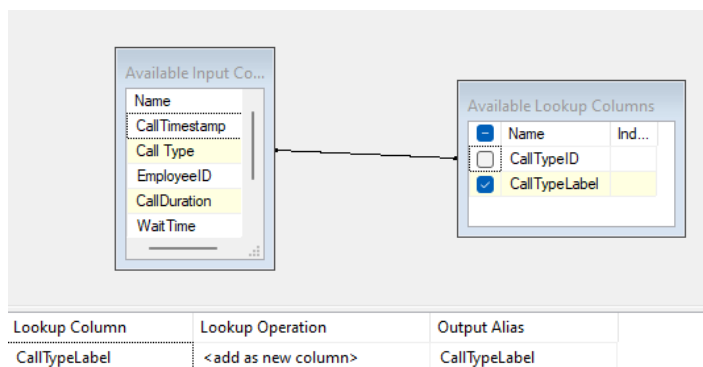
<for now, there is no error>

```
1 CREATE TABLE [Technical_Rejects] (
2     [RejectDate] datetime,
3     [RejectPackageAndTask] nvarchar(203),
4     [RejectColumn] nvarchar(20),
5     [RejectDescription] nvarchar(129)
6 )
```

<Creation of "Technical\_Rejects" table in ADM SQL Server Database>

## Part 2 : Join the "Call Types" table with "Lookup" SSIS component.

- **LKP - Call Types Label** : Drag and drop "Lookup" SSIS component in "Data Flow" → Connect to the SQL Server database where the STA Call Types table is stored. → Select the CallType table → Go to the Columns menu then link between Call Type and CallTypeID columns then mark the check box of CallTypeLabel.



## - Data Viewer

CallAbandoned	CallTimestamp_date	CallDuration_int	CallAbandoned_bool	CallTypeLabel
0	2018-04-05 16:33:00.0000000	486	False	Tech Support
0	2018-06-21 18:28:00.0000000	945	False	Tech Support
0	2018-06-21 15:13:00.0000000	379	False	Sales
0	2018-11-21 13:02:00.0000000	1044	False	Tech Support
0	2018-02-25 13:36:00.0000000	1357	False	Sales
0	2018-10-28 10:04:00.0000000	570	False	Tech Support
0	2018-12-17 16:39:00.0000000	26	False	Tech Support
0	2018-07-28 19:09:00.0000000	8	False	Billing
0	2018-06-11 18:57:00.0000000	800	False	Sales
0	2018-06-18 15:32:00.0000000	651	False	Billing
0	2018-05-14 16:24:00.0000000	229	False	Billing
0	2018-02-06 12:57:00.0000000	467	False	Tech Support
0	2018-06-20 12:01:00.0000000	1290	False	Tech Support
0	2018-12-20 16:27:00.0000000	504	False	Tech Support
0	2018-12-15 16:23:00.0000000	1445	False	Tech Support

Attached Total rows: 0, buffers: 0 Rows displayed = 2110



### Part 3 : Join the “Call Charges” table with “Call Data” table

“CallChargeID” will be constituted by the “Year” (from CallTimestamp) and the first alphabet of the “CallTypeLabel”. For example, 2018B. For combining these two elements, the LEFT() function will be used. Currently, at this phase, the “CallTimestamp\_date” derived column has format “dd/mm/yyyy hh:ss.ssssss”. For the LEFT() function, the date format should be “YYY/DD/MM” then, the statement will be “LEFT(CallDate,4) + LEFT(CallTypeLabel,1)” which combines these two elements by slicing 4 letter from CallDate and 1 letter from CallTypeLabel from the left side.

- **Separate the CallTimestamp into CallDate and CallTime.**  
First, arrangement of date format will be applied.
- **DER - CallDate and CallTime :** Select Lookup Match Output → Select CallTimestamp in “Available Input Columns”  
→ Derived Column Name : CallDate , Derived Column : <add as new column>,  
Expression : (DT\_DBDATE)CallTimestamp\_date  
→ Derived Column Name : CallTime , Derived Column : <add as new column>,  
Expression : (DT\_DBTIME2,0)CallTimestamp\_date  
- Data Viewer

CallTimestamp	Call Type	EmployeeID	CallDuration	WaitTime	CallAbandoned	CallTimestamp_date	Call...	Wai...	C...	C...	CallTypeLabel	CallDate	CallTime
1/1/2018 13:34	3	C206355	285	110	0	2018-01-01 13:34:00.0000000	285	110	3	F...	Tech Support	01/01/2018	13:34:00
1/1/2018 9:25	2	F533051	175	17	0	2018-01-01 09:25:00.0000000	175	17	2	F...	Billing	01/01/2018	09:25:00
1/1/2018 12:54	2	B651033	903	22	0	2018-01-01 12:54:00.0000000	903	22	2	F...	Billing	01/01/2018	12:54:00
1/1/2018 12:48	2	K915074	552	25	1	2018-01-01 12:48:00.0000000	552	25	2	T...	Billing	01/01/2018	12:48:00
1/1/2018 11:12	3	I543040	1033	0	0	2018-01-01 11:12:00.0000000	1033	0	3	F...	Tech Support	01/01/2018	11:12:00
1/1/2018 14:01	2	K399501	640	0	0	2018-01-01 14:01:00.0000000	640	0	2	F...	Billing	01/01/2018	14:01:00
1/1/2018 10:21	2	U641256	416	18	0	2018-01-01 10:21:00.0000000	416	18	2	F...	Billing	01/01/2018	10:21:00
1/1/2018 18:38	2	S704443	1127	0	1	2018-01-01 18:38:00.0000000	1127	0	2	T...	Billing	01/01/2018	18:38:00
2/1/2018 9:06	3	B651033	1314	20	0	2018-01-02 09:06:00.0000000	1314	20	3	F...	Tech Support	02/01/2018	09:06:00
2/1/2018 17:49	3	P286634	113	26	0	2018-01-02 17:49:00.0000000	113	26	3	F...	Tech Support	02/01/2018	17:49:00
3/1/2018 17:30	2	K669566	1195	20	0	2018-01-03 17:30:00.0000000	1195	20	2	F...	Billing	03/01/2018	17:30:00
3/1/2018 16:50	3	K222775	243	27	0	2018-01-03 16:50:00.0000000	243	27	3	F...	Tech Support	03/01/2018	16:50:00
3/1/2018 15:42	3	O856292	129	13	0	2018-01-03 15:42:00.0000000	129	13	3	F...	Tech Support	03/01/2018	15:42:00
3/1/2018 15:37	3	V126561	389	141	1	2018-01-03 15:37:00.0000000	389	141	3	T...	Tech Support	03/01/2018	15:37:00

Attached Total rows: 0, buffers: 0 Rows displayed = 1867

<the complexity of datetime format is solved>

- **Create a new CallChargeID and join the Call Data and Call Charges tables.**  
As the LEFT() function requires a string data type, at first, cast CallDate into string data type then create a new CallChargeID column.
- **DCNV - CallDate Str :** Using the “Data Conversion” SSIS component, cast the data type of “CallDate” into a strig. As with the other data conventions, use the “Derived

Column” SSIS component for the rejects and direct the error values to “OLE\_DST - AMD Rejects”

- **DER - CallChargeID** : Derived Column Name : CallChargeID, Derived Column : <add as new column>, Expression : (DT\_DBDATE) LEFT(CallDate\_str,4) + LEFT(CallTypeLabel,1)

Derived Column Output Data Viewer at DFT - ODS Call Data

Detach Copy Data

CallTimestamp	Call Type	EmployeeID	CallDuration	WaitTime	CallAbandoned	CallTimestamp_date	CallDuration_int	CallAbandoned_bool	CallTypeLabel	CallDate	CallTime	CallDate_str	CallChargeID
5/4/2018 16:33	3	U559430	486	2	0	2018-04-05 16:33:00.0000000	486	False	Tech Support	05/04/2018	16:33:00	2018-04-05	2018T
6/21/2018 18:28	3	A166733	945	0	0	2018-06-21 18:28:00.0000000	945	False	Tech Support	21/06/2018	18:28:00	2018-06-21	2018T
6/21/2018 15:13	1	B971624	379	11	0	2018-06-21 15:13:00.0000000	379	False	Sales	21/06/2018	15:13:00	2018-06-21	2018S
11/21/2018 13:02	3	U641256	1044	0	0	2018-11-21 13:02:00.0000000	1044	False	Tech Support	21/11/2018	13:02:00	2018-11-21	2018T
2/25/2018 13:36	1	P286634	1357	0	0	2018-02-25 13:36:00.0000000	1357	False	Sales	25/02/2018	13:36:00	2018-02-25	2018S
10/28/2018 10:04	3	M855788	570	23	0	2018-10-28 10:04:00.0000000	570	False	Tech Support	28/10/2018	10:04:00	2018-10-28	2018T
12/17/2018 16:39	3	M794992	26	26	0	2018-12-17 16:39:00.0000000	26	False	Tech Support	17/12/2018	16:39:00	2018-12-17	2018T
7/28/2018 19:09	2	I281837	8	8	0	2018-07-28 19:09:00.0000000	8	False	Billing	28/07/2018	19:09:00	2018-07-28	2018B
11/6/2018 18:57	1	J192194	800	0	0	2018-06-11 18:57:00.0000000	800	False	Sales	11/06/2018	18:57:00	2018-06-11	2018S
6/18/2018 15:32	2	F542348	651	111	0	2018-06-18 15:32:00.0000000	651	False	Billing	18/06/2018	15:32:00	2018-06-18	2018B
5/14/2018 16:24	2	J632516	229	10	0	2018-05-14 16:24:00.0000000	229	False	Billing	14/05/2018	16:24:00	2018-05-14	2018B
6/2/2018 12:57	3	U194381	467	23	0	2018-02-06 12:57:00.0000000	467	False	Tech Support	06/02/2018	12:57:00	2018-02-06	2018T
6/20/2018 12:01	3	G586239	1290	7	0	2018-06-20 12:01:00.0000000	1290	False	Tech Support	20/06/2018	12:01:00	2018-06-20	2018T
12/20/2018 16:27	3	N772493	504	0	0	2018-12-20 16:27:00.0000000	504	False	Tech Support	20/12/2018	16:27:00	2018-12-20	2018T
12/15/2018 16:23	3	J632516	1445	28	0	2018-12-15 16:23:00.0000000	1445	False	Tech Support	15/12/2018	16:23:00	2018-12-15	2018T
3/4/2018 15:34	2	K669566	1255	2	0	2018-04-03 15:34:00.0000000	1255	False	Billing	03/04/2018	15:34:00	2018-04-03	2018B
12/23/2018 14:03	3	E137708	1069	5	0	2018-12-23 14:03:00.0000000	1069	False	Tech Support	23/12/2018	14:03:00	2018-12-23	2018T
4/17/2018 8:00	3	M855788	758	0	0	2018-04-17 08:00:00.0000000	758	False	Tech Support	17/04/2018	08:00:00	2018-04-17	2018T
2/11/2018 8:39	2	E778362	838	0	0	2018-11-02 08:39:00.0000000	838	False	Billing	02/11/2018	08:39:00	2018-11-02	2018B
11/22/2018 14:44	3	N119221	681	10	0	2018-11-22 14:44:00.0000000	681	False	Tech Support	22/11/2018	14:44:00	2018-11-22	2018T
8/4/2018 11:51	2	V228277	1272	42	0	2018-04-08 11:51:00.0000000	1272	False	Billing	08/04/2018	11:51:00	2018-04-08	2018B
7/11/2018 14:34	3	T398672	252	0	0	2018-11-07 14:34:00.0000000	252	False	Tech Support	07/11/2018	14:34:00	2018-11-07	2018T
11/24/2018 11:48	3	Q921541	84	271	0	2018-11-24 11:48:00.0000000	84	False	Tech Support	24/11/2018	11:48:00	2018-11-24	2018T
3/22/2018 11:37	2	E243130	229	132	0	2018-03-22 11:37:00.0000000	229	False	Billing	22/03/2018	11:37:00	2018-03-22	2018B

Attached Total rows: 0, buffers: 0 Rows displayed = 2120

- **LKP - Call Charges** : Select Lookup Match Output → Connect to ODS SQL server database and select the “CallCharges” table → Go to Columns menu → Link CallChargeId and CallChargeID then check the box for Call Charges

General  
Connection  
Columns  
Advanced  
Error Output

Available Input Columns

Name
CallTimestamp
Call Type
EmployeeID
CallDuration
WaitTime
CallAbandoned
CallTimestamp_date
CallDuration_int
CallAbandoned_bool
CallTypeLabel
CallDate
CallTime
CallDate_str
CallChargeID

Available Lookup Columns

Name	Ind...
<input type="checkbox"/> CallChargeID	
<input checked="" type="checkbox"/> Call Charges	
<input type="checkbox"/> Call Type	
<input type="checkbox"/> Year	

Lookup Match Output Data Viewer at DFT - ODS Call Data

CallTime...	Call...	C...	CallTypeLabel	CallDate	CallTime	CallDate_str	CallChargeID	Call Charges
2018-0...	486	F...	Tech Support	05/04/...	16:33:00	2018-04-05	2018T	0.95 / min
2018-0...	945	F...	Tech Support	21/06/...	18:28:00	2018-06-21	2018T	0.95 / min
2018-0...	379	F...	Sales	21/06/...	15:13:00	2018-06-21	2018S	1.52 / min
2018-1...	1044	F...	Tech Support	21/11/...	13:02:00	2018-11-21	2018T	0.95 / min
2018-0...	1357	F...	Sales	25/02/...	13:36:00	2018-02-25	2018S	1.52 / min
2018-1...	570	F...	Tech Support	28/10/...	10:04:00	2018-10-28	2018T	0.95 / min
2018-1...	26	F...	Tech Support	17/12/...	16:39:00	2018-12-17	2018T	0.95 / min
2018-0...	8	F...	Billing	28/07/...	19:09:00	2018-07-28	2018B	1.2 / min
2018-0...	800	F...	Sales	11/06/...	18:57:00	2018-06-11	2018S	1.52 / min
2018-0...	651	F...	Billing	18/06/...	15:32:00	2018-06-18	2018B	1.2 / min
2018-0...	229	F...	Billing	14/05/...	16:24:00	2018-05-14	2018B	1.2 / min
2018-0...	467	F...	Tech Support	06/02/...	12:57:00	2018-02-06	2018T	0.95 / min
2018-0...	1290	F...	Tech Support	20/06/...	12:01:00	2018-06-20	2018T	0.95 / min
2018-1...	504	F...	Tech Support	20/12/...	16:27:00	2018-12-20	2018T	0.95 / min
2018-1...	1445	F...	Tech Support	15/12/...	16:23:00	2018-12-15	2018T	0.95 / min

Attached Total rows: 0, buffers: 0 Rows displayed = 2110

<Data Viewer>

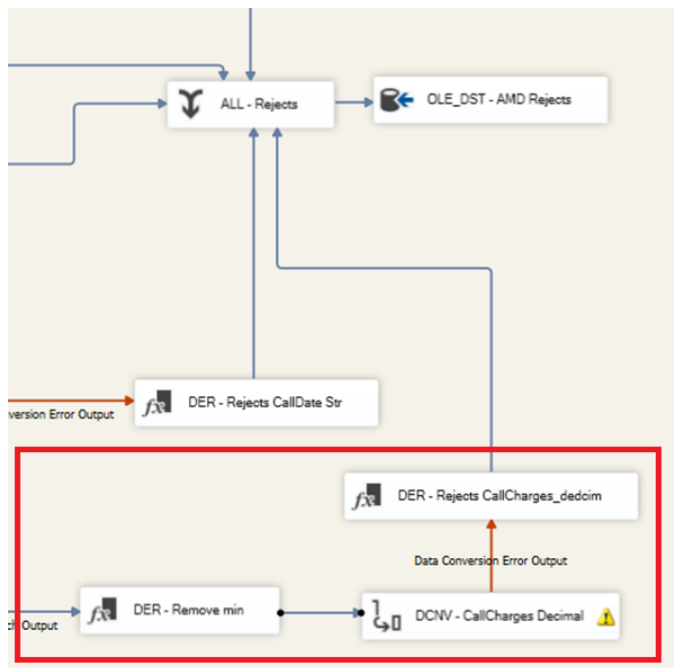
- **DER - Remove min** : Derived Column Name : CallCharge\_remove\_min, Derived Column : <add as new column>, Expression : (DT\_DBDATE) LEFT(CallDate\_str,4) + LEFT(CallTypeLabel,1)

Derived Column Output Data Viewer at DFT - ODS Call Data

CallTypeLabel	CallDate	CallTime	CallDate_str	CallChargeID	Call Charges	CallCharges_remove_min
Tech Support	05/04/...	16:33:00	2018-04-05	2018T	0.95 / min	0.95
Tech Support	21/06/...	18:28:00	2018-06-21	2018T	0.95 / min	0.95
Sales	21/06/...	15:13:00	2018-06-21	2018S	1.52 / min	1.52
Tech Support	21/11/...	13:02:00	2018-11-21	2018T	0.95 / min	0.95
Sales	25/02/...	13:36:00	2018-02-25	2018S	1.52 / min	1.52
Tech Support	28/10/...	10:04:00	2018-10-28	2018T	0.95 / min	0.95
Tech Support	17/12/...	16:39:00	2018-12-17	2018T	0.95 / min	0.95
Billing	28/07/...	19:09:00	2018-07-28	2018B	1.2 / min	1.2
Sales	11/06/...	18:57:00	2018-06-11	2018S	1.52 / min	1.52
Billing	18/06/...	15:32:00	2018-06-18	2018B	1.2 / min	1.2
Billing	14/05/...	16:24:00	2018-05-14	2018B	1.2 / min	1.2
Tech Support	06/02/...	12:57:00	2018-02-06	2018T	0.95 / min	0.95
Tech Support	20/06/...	12:01:00	2018-06-20	2018T	0.95 / min	0.95
Tech Support	20/12/...	16:27:00	2018-12-20	2018T	0.95 / min	0.95

Attached Total rows: 0, buffers: 0 Rows displayed = 1822

- **DCNV - CallCharges Decimal** : Cast data type into decimal then iterate same process for the “ADM Rejects”



#### Part 4 : SLA Status

- **DER - SLA Status** : Derived Column Name : SLA Status, Derived Column : <add as new column>, Expression : `WaitTime_int < 35 ? "Within SLA" : "Outside SLA"`

Derived Column Output Data Viewer at DFT - ODS Call Data

Detach Copy Data

CallDate	CallTime	CallDate_str	CallChargeID	Call Charges	CallCharges_remove_...	CallChar...	SLA Status
05/04/...	16:33:00	2018-04-05	2018T	0.95 / min	0.95	0	Within SLA
21/06/...	18:28:00	2018-06-21	2018T	0.95 / min	0.95	0	Within SLA
21/06/...	15:13:00	2018-06-21	2018S	1.52 / min	1.52	1	Within SLA
21/11/...	13:02:00	2018-11-21	2018T	0.95 / min	0.95	0	Within SLA
25/02/...	13:36:00	2018-02-25	2018S	1.52 / min	1.52	1	Within SLA
28/10/...	10:04:00	2018-10-28	2018T	0.95 / min	0.95	0	Within SLA
17/12/...	16:39:00	2018-12-17	2018T	0.95 / min	0.95	0	Within SLA
28/07/...	19:09:00	2018-07-28	2018B	1.2 / min	1.2	1	Within SLA
11/06/...	18:57:00	2018-06-11	2018S	1.52 / min	1.52	1	Within SLA
18/06/...	15:32:00	2018-06-18	2018B	1.2 / min	1.2	1	Outside ...
14/05/...	16:24:00	2018-05-14	2018B	1.2 / min	1.2	1	Within SLA
06/02/...	12:57:00	2018-02-06	2018T	0.95 / min	0.95	0	Within SLA
20/06/...	12:01:00	2018-06-20	2018T	0.95 / min	0.95	0	Within SLA
20/12/...	16:27:00	2018-12-20	2018T	0.95 / min	0.95	0	Within SLA

Attached Total rows: 0, buffers: 0 Rows displayed = 1598

## Part 5 : ODS CallData

- DCNV - Resizing : Reconfigure the length

The 'Available Input Columns' dialog box shows a list of columns with checkboxes. The columns are: Name (checked), WaitTime\_int, WaitTime, SLA Status, EmployeeID (checked), CallTypeLabel (checked), CallTimestamp..., CallTimestamp, and CallTime.

Input Column	Output Alias	Data Type	Length	Precision	Scale	Code Page
EmployeeID	EmployeeID_R	string [DT_STR]	10			1252 (ANSI - Latin I)
CallTypeLabel	CallTypeLabel_R	string [DT_STR]	20			1252 (ANSI - Latin I)

- OLE\_DST - ODS CallData : Connect to SQL Server ODS database then select the "CallData" table and create a new "CallData" table → Mapping

```

12 CREATE TABLE [CallData] (
13     [EmployeeID] varchar(10),
14     [CallDate] date,
15     [CallTime] time(0),
16     [CallType] varchar(20),
17     [CallDuration] int,
18     [CallCharges] decimal(28,2),
19     [WaitTime] int,
20     [SLA Status] nvarchar(15),
21     [CallAbandoned] bit
22 )

```

The screenshot shows the 'Available Input Columns' and 'Available Destination Columns' lists. The 'Available Input Columns' list includes: Name, CallAbandoned, CallTimestamp\_date, WaitTime\_int, CallDuration\_int, CallAbandoned\_bool, CallTypeLabel, CallDate, CallTime, CallDate\_str, CallChargeID, CallCharges\_remove\_min, CallCharges\_decim, SLA Status, EmployeeID\_R, and CallTypeLabel\_R. The 'Available Destination Columns' list includes: Name, EmployeeID, CallDate, CallTime, CallDuration, WaitTime, SLA Status, CallAbandoned, CallType, and CallCharges. Lines connect the input columns to the destination columns: EmployeeID\_R to EmployeeID, CallDate to CallDate, CallTime to CallTime, CallDuration\_int to CallDuration, WaitTime\_int to WaitTime, SLA Status to SLA Status, CallAbandoned\_bool to CallAbandoned, CallTypeLabel\_R to CallType, and CallCharges\_decim to CallCharges.

Input Column	Destination Column
EmployeeID_R	EmployeeID
CallDate	CallDate
CallTime	CallTime
CallDuration_int	CallDuration
WaitTime_int	WaitTime
SLA Status	SLA Status
CallAbandoned_bool	CallAbandoned
CallTypeLabel_R	CallType
CallCharges_decim	CallCharges

## Result “CallData” table :

	EmployeeID	CallDate	CallTime	CallType	CallDuration	CallCharges	WaitTime	SLA Status	CallAbandoned
1	F542348	2020-01-20	16:12:00	Billing	90	1.41	2	Within SLA	0
2	K273930	2020-10-06	19:18:00	Tech Support	342	1.04	8	Within SLA	0
3	N470942	2020-01-07	16:55:00	Billing	278	1.41	0	Within SLA	0
4	G586239	2020-10-24	11:42:00	Tech Support	1282	1.04	0	Within SLA	0
5	I992869	2020-10-19	12:57:00	Tech Support	697	1.04	0	Within SLA	0
6	C206355	2020-08-14	09:12:00	Tech Support	677	1.04	0	Within SLA	0
7	O362956	2020-06-13	10:00:00	Tech Support	1453	1.04	0	Within SLA	0
8	J173690	2020-12-05	19:04:00	Billing	270	1.41	0	Within SLA	0
9	N470942	2020-07-26	12:45:00	Tech Support	623	1.04	3	Within SLA	0
10	T387561	2020-06-20	17:36:00	Tech Support	745	1.04	125	Outside SLA	0
11	V828034	2020-11-01	19:27:00	Tech Support	1026	1.04	25	Within SLA	0
12	T387561	2020-01-21	16:01:00	Billing	27	1.41	13	Within SLA	0
13	K273930	2020-05-29	17:43:00	Sales	1352	1.60	6	Within SLA	0
14	U559430	2020-08-11	10:01:00	Tech Support	1339	1.04	17	Within SLA	0
15	G727038	2020-05-07	09:06:00	Billing	1302	1.41	22	Within SLA	0
16	N772493	2020-08-21	17:34:00	Tech Support	323	1.04	15	Within SLA	0
17	S564705	2020-06-23	18:03:00	Billing	167	1.41	0	Within SLA	0
18	M794992	2020-02-23	09:49:00	Tech Support	581	1.04	0	Within SLA	0
19	V228277	2020-09-03	13:19:00	Tech Support	377	1.04	0	Within SLA	0
20	J632516	2020-02-23	12:10:00	Sales	298	1.60	8	Within SLA	0
21	T398672	2020-05-16	17:26:00	Tech Support	76	1.04	22	Within SLA	0
22	U641256	2020-11-13	09:49:00	Billing	1025	1.41	0	Within SLA	0
23	D774655	2020-07-18	11:43:00	Billing	937	1.41	0	Within SLA	0

Query executed successfully. | MSI (15.0 RTM) | AzureAD\EvanKim (56) | Report\_CallCenter\_ODS | 00:00:01 | 197,950 rows

## III-3. Employees Table

The Site column has a name of site and a State Code both. Dividing by State Code and the Site, State code will be joined with the US States table to have correct name of State and Region.

- Verify if there is any duplicate data

```

1 SELECT [EmployeeID]
2       , [EmployeeName]
3       , [Site]
4       , [ManagerName]
5       , COUNT (*) AS DuplicateNB
6 FROM Employees
7
8
9 GROUP BY [EmployeeID]
10         , [EmployeeName]
11         , [Site]
12         , [ManagerName]
13 HAVING COUNT (*) > 1
14

```

108 %

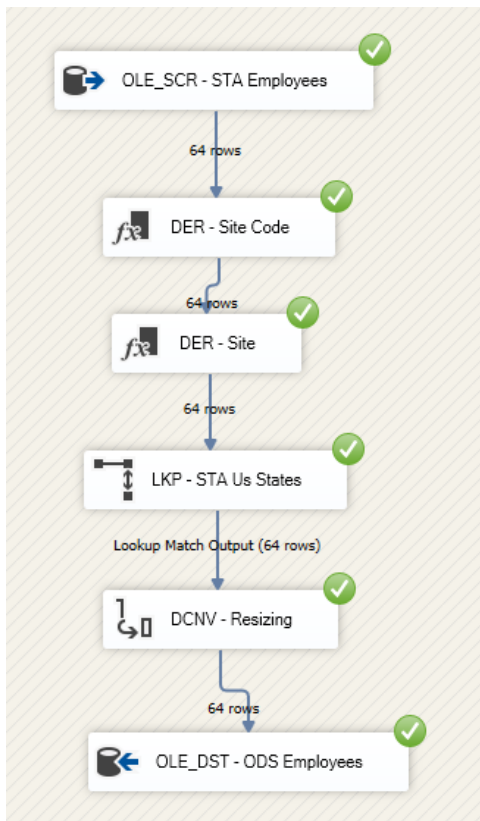
Results Messages

EmployeeID	EmployeeName	Site	ManagerName	DuplicateNB
------------	--------------	------	-------------	-------------

<No duplicate>

- Data Flow for the ODS Employees data





- DER - Site Code :

Derived Column Name	Derived Column	Expression	Data Type	Length
SiteCode	<add as new column>	(DT_STR,10,1252)RIGHT(Site,2)	string [DT_STR]	10

- DER - Site :

Derived Column Name	Derived Column	Expression	Data Type	Length
Site	<add as new column>	SUBSTRING(Site,1,LEN(Site) - 4)	Unicode string [DT_WS...	255

- LKP - STA Us States :

General  
Connection  
Columns  
Advanced  
Error Output

Available Input Columns

Name
OLE_SCR - STA Employees.Site
ManagerName
SiteCode
DER - Site.Site

Available Lookup Colu...

<input checked="" type="checkbox"/>	Name	Ind...
<input type="checkbox"/>	StateCD	
<input checked="" type="checkbox"/>	Name	
<input checked="" type="checkbox"/>	Region	

Lookup Column	Lookup Operation	Output Alias
Name	<add as new column>	State
Region	<add as new column>	Region

- DCNV - Resizing :

Input Column	Output Alias	Data Type	Length	Precision	Scale	Code Page
EmployeeID	EmployeeID_R	string [DT_STR]	30			1252 (ANSI - Latin I)
EmployeeName	EmployeeName_R	string [DT_STR]	50			1252 (ANSI - Latin I)
ManagerName	ManagerName_R	string [DT_STR]	50			1252 (ANSI - Latin I)
DER - Site.Site	Site_R	Unicode string [DT_WSTR]	30			
State	State_R	string [DT_STR]	30			1252 (ANSI - Latin I)
Region	Region_R	string [DT_STR]	30			1252 (ANSI - Latin I)

- OLE\_DST - ODS Employees :

```

28 CREATE TABLE [Employees] (
29     [EmployeeID] varchar(30),
30     [EmployeeName] varchar(50),
31     [ManagerName] varchar(50),
32     [Site] nvarchar(30),
33     [State] varchar(30),
34     [Region] varchar(30)
35 )

```

	EmployeeID	EmployeeName	ManagerName	Site	State	Region
1	N72493	Onita Trojan	Deidre Robbs	Spokane	Washington	West
2	F533051	Stormy Seller	Elsie Taplin	Aurora	Colorado	West
3	S564705	Mable Ayoub	Shala Lion	Aurora	Colorado	West
4	I281837	Latisha Buckalew	Rana Taub	Aurora	Colorado	West
5	Y193775	Adrianna Duque	Collin Trotman	Spokane	Washington	West
6	J632516	Keiko Daulton	Jamar Prael	Spokane	Washington	West
7	G727038	Dolores Lundeen	Shala Lion	Aurora	Colorado	West
8	V126561	Wilbur Mohl	Casey Bainbridge	Jacksonville	Florida	South
9	E243130	Ileen Bomstein	Gonzalo Lesage	Jacksonville	Florida	South
10	C206355	Janeth Roesler	Miyoko Degraw	Spokane	Washington	West
11	G586239	Shery Hover	Elsie Taplin	Aurora	Colorado	West
12	M163408	Trevor Cerda	Collin Trotman	Spokane	Washington	West
13	H438047	Debora Wilker	Collin Trotman	Spokane	Washington	West
14	T659609	Kelvin Nicoletti	Jamar Prael	Spokane	Washington	West
15	P286634	Rafael Langton	Elsie Taplin	Aurora	Colorado	West
16	T387561	Alla Winkel	Ardath Ducharme	Jacksonville	Florida	South
17	J419954	Harrison Finlayson	Rana Taub	Aurora	Colorado	West
18	N470942	Caterina Jantz	Miyoko Degraw	Spokane	Washington	West
19	B971624	Agripina Snively	Elsie Taplin	Aurora	Colorado	West
20	O856292	Rosenda Korus	Gonzalo Lesage	Jacksonville	Florida	South
21	T664783	Ivette Rodenberg	Gonzalo Lesage	Jacksonville	Florida	South
22	U194381	Beulah Aubert	Ardath Ducharme	Jacksonville	Florida	South
23	N119221	Aleida Singh	Elsie Taplin	Aurora	Colorado	West

Query executed succes... | MSI (15.0 RTM) | AzureAD\EvanKim (62) | Report\_CallCenter\_ODS | 00:00:00 | 64 rows



## IV. DATA WARE HOUSE

Beyond of the ODS phase, It should be ensured the quality of the data before working on the Data Ware Housing. First, add a Surrogate Key for each Dimension table in order to verify if there is missing value or errors. Second, build certain use case based on result of ETL tasks.

### IV-1. Date Dimension Table

- Create a Date Dimension Table based on the Dim Date code on SQL Server.

### IV-2. Employees Dimension Table

- **OLE\_SRC - ODS Employees** : Connect to the ODS SQL Server database then load ODS Employees table.
- **LKP - EmployeeID** : General menu → Select “Redirect rows to no match output” → Connection menu → Create a new “DimEmployees” table

```
USE [Report_CallCenter_DWH]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[DimEmployees](
    [EmployeeSurrKey] INT PRIMARY KEY IDENTITY (1,1),
    [EmployeeID] [varchar](30) NULL,
    [EmployeeName] [varchar](50) NULL,
    [ManagerName] [varchar](50) NULL,
    [Site] [nvarchar](30) NULL,
    [State] [varchar](30) NULL,
    [Region] [varchar](30) NULL
) ON [PRIMARY]
GO
```

→ Select the DWH SQL Server database and “DimEmployees” table → Only link the data between the “EmployeeID” and “EmployeeID” columns.

- “INT PRIMARY KEY Identity (1,1)” will be executed to create a new “DimEmployees” table. It incrementally generates the surrogate key of Employees data one by one according to (1,1) command.
- The “Redirect rows to no match output” setting redirect unmatched rows referring to the EmployeeID.
- **LKP - Any Change?** : Link “LKP - EmployeeID” into “LKP - Any Change?” then select “Lookup Match Output” → General menu → Select “Redirect rows to no match output” → Connection menu → check the match columns except Input Column “EmployeeSurrKey” this column will get the Surrogate Key by previous scrip table to have created the “DimEmployees” table
- **CMD - Update Record** : Link “LKP - Any Change?” into “CMD - Update Record” and select Lookup Match Output → Connection Managers tap → Select the ODS SQL Server database → Component Properties tap → Write a script to update unmatched data

String Value Editor

String value:

```
UPDATE [dbo].[Employees]
SET
    [EmployeeName] = ?
    ,[ManagerName] = ?
    ,[Site] = ?
    ,[State] = ?
    ,[Region] = ?
WHERE [EmployeeID] = ?
```

→ Mapp the columns

Connection Managers Component Properties Column Mappings Input and Output Properties

Available Input Columns	Available Destination...
Name	Name
EmployeeID	Param_0
EmployeeName	Param_1
ManagerName	Param_2
Site	Param_3
State	Param_4
Region	Param_5

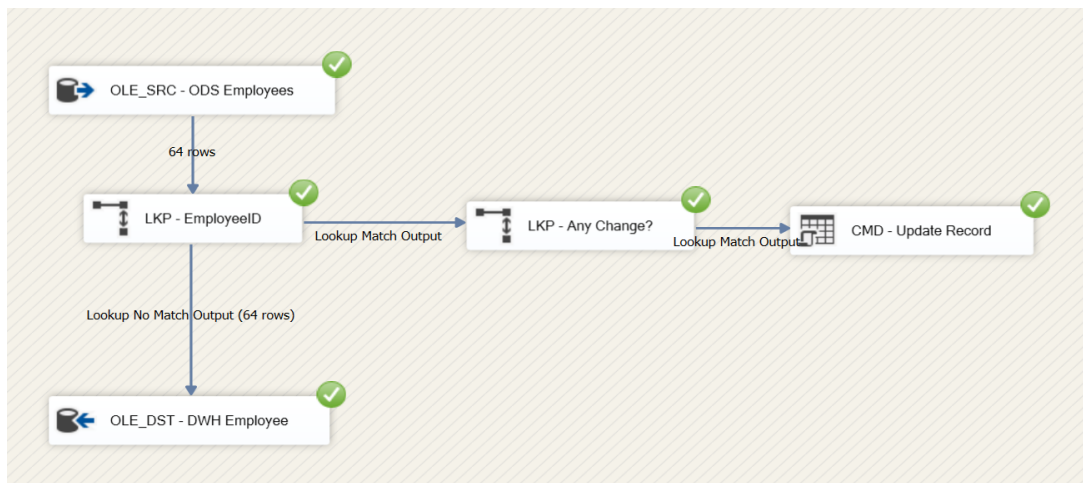
  

Input Column	Destination Column
EmployeeName	Param_0
ManagerName	Param_1
Site	Param_2
State	Param_3
Region	Param_4
EmployeeID	Param_5

- The Destination Column follows the order of generated table by “String Value

Editor”.

- **OLE\_DST - DWH Employee** : Link “LKP - EmployeeID” into “OLE\_DST - DWH Employee” then select Lookup No Match Output.



	EmployeeSurKey	EmployeeID	EmployeeName	ManagerName	Site	State	Region
1	1	N772493	Onita Trojan	Deidre Robbs	Spokane	Washington	West
2	2	F533051	Stormy Seller	Elsie Taplin	Aurora	Colorado	West
3	3	S564705	Mable Ayoub	Shala Lion	Aurora	Colorado	West
4	4	I281837	Latrisha Buckalew	Rana Taub	Aurora	Colorado	West
5	5	Y193775	Adrianna Duque	Collin Trotman	Spokane	Washington	West
6	6	J632516	Keiko Daulton	Jamar Pahl	Spokane	Washington	West
7	7	G727038	Dolores Lundeen	Shala Lion	Aurora	Colorado	West
8	8	V126561	Wilbur Mohl	Casey Bainbridge	Jacksonville	Florida	South
9	9	E243130	Ileen Bomstein	Gonzalo Lesage	Jacksonville	Florida	South
10	10	C206355	Janeth Roesler	Miyoko Degraw	Spokane	Washington	West
11	11	G586239	Shery Hover	Elsie Taplin	Aurora	Colorado	West
12	12	M163408	Trevor Cerda	Collin Trotman	Spokane	Washington	West
13	13	H438047	Debora Wilker	Collin Trotman	Spokane	Washington	West
14	14	T659609	Kelvin Nicoletti	Jamar Pahl	Spokane	Washington	West
15	15	P286634	Rafael Langton	Elsie Taplin	Aurora	Colorado	West
16	16	T387561	Alla Winkel	Ardath Duchame	Jacksonville	Florida	South
17	17	J419954	Hamison Finlayson	Rana Taub	Aurora	Colorado	West
18	18	N470942	Caterina Jantz	Miyoko Degraw	Spokane	Washington	West
19	19	B971624	Agripina Snively	Elsie Taplin	Aurora	Colorado	West
20	20	O856292	Rosenda Korus	Gonzalo Lesage	Jacksonville	Florida	South

Query executed successfully. MSI (15.0 RTM) AzureAD\EvanKim (66) Report\_CallCenter\_DWH 00:00:00 64 rows

- Now the Dimension Employees table is ready to be using for some business purposes of analysis. Forming a “Star Schema” , it will be deployed with the Fact Table “Call Data”

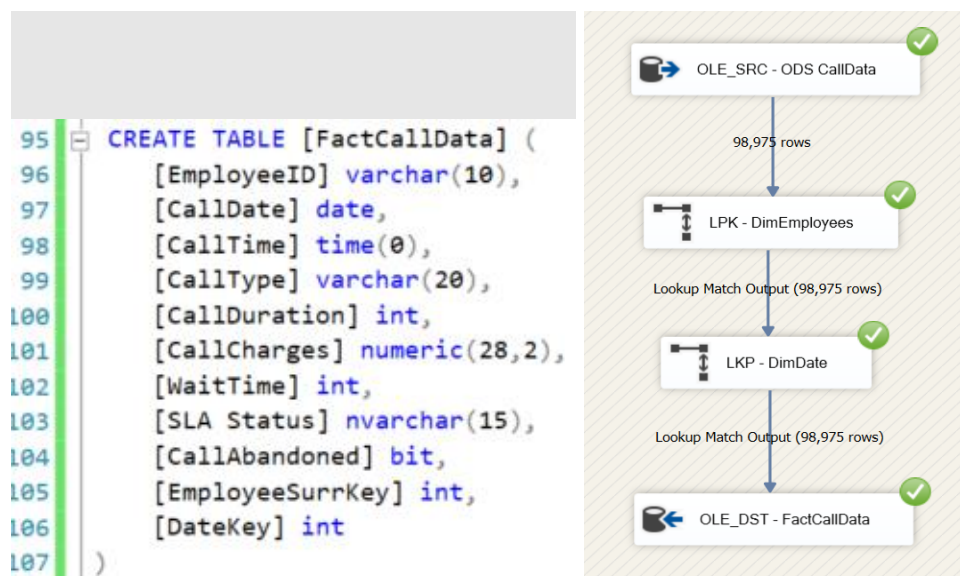
### IV-3. Call Data Fact Table

For ensuring the quality of data on the Fact table, deployed SCD type 1 method. It calls “Slowly Changing Dimension”. It overwrites old data(Primary Key) with the new data(Surrogate Key) in dimension table.

- Join these two table “DimEmployees” and “FactCallData” with a Primary key “EmployeeID” and “EmployeeSurrogateKey”
- Check the missing values from the EmployeeID
- Make the outputs in case of the missing value and unmatched values referring this Primary key “EmployeeID”

#### Part 1. Lookup EmployeeID and DateKey

- OLE\_SRC - ODS CallData : Connect to the SQL Server database ODS CallData then load ODS CallData table.
- LPK - DimEmployees : make a relationship between “DimEmployees” table and “FactCallData” table.
- LKP - DimDate : make a relationship between “DimDate” table and “FactCallData” table. With “Date” column and “CallDate” column .
- OLE\_DST - FactCallData : Before preceding farther, observe the first table of “FacCammData”.



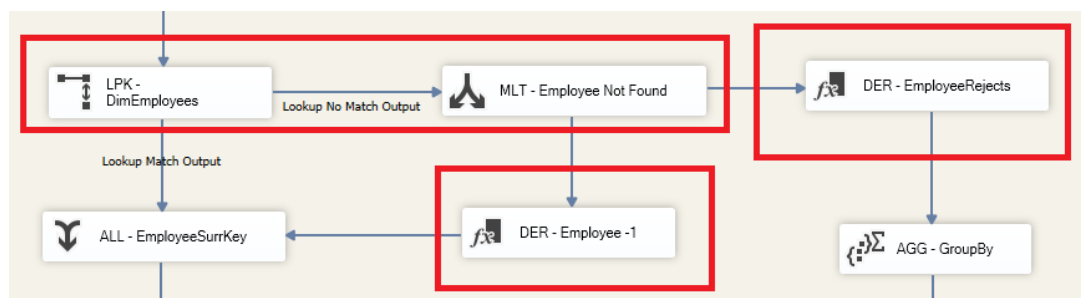
	EmployeeID	CallDate	CallTime	CallType	CallDuration	CallCharges	WaitTime	SLA Status	CallAbandoned	EmployeeSurrKey	DateKey
1	U559430	2018-04-05	16:33:00	Tech Support	486	0.95	2	Within SLA	0	45	20180405
2	A166733	2018-06-21	18:28:00	Tech Support	945	0.95	0	Within SLA	0	38	20180621
3	B971624	2018-06-21	15:13:00	Sales	379	1.52	11	Within SLA	0	19	20180621
4	U641256	2018-11-21	13:02:00	Tech Support	1044	0.95	0	Within SLA	0	36	20181121
5	P286634	2018-02-25	13:36:00	Sales	1357	1.52	0	Within SLA	0	15	20180225
6	M855788	2018-10-28	10:04:00	Tech Support	570	0.95	23	Within SLA	0	24	20181028
7	M794992	2018-12-17	16:39:00	Tech Support	26	0.95	26	Within SLA	0	33	20181217
8	I281837	2018-07-28	19:09:00	Billing	8	1.20	8	Within SLA	0	4	20180728
9	J192194	2018-06-11	18:57:00	Sales	800	1.52	0	Within SLA	0	54	20180611
10	F542348	2018-06-18	15:32:00	Billing	651	1.20	111	Outside SLA	0	61	20180618
11	J632516	2018-05-14	16:24:00	Billing	229	1.20	10	Within SLA	0	6	20180514
12	U194381	2018-02-06	12:57:00	Tech Support	467	0.95	23	Within SLA	0	22	20180206
13	G586239	2018-06-20	12:01:00	Tech Support	1290	0.95	7	Within SLA	0	11	20180620
14	N772493	2018-12-20	16:27:00	Tech Support	504	0.95	0	Within SLA	0	1	20181220
15	J632516	2018-12-15	16:23:00	Tech Support	1445	0.95	28	Within SLA	0	6	20181215
16	K669566	2018-04-03	15:34:00	Billing	1255	1.20	2	Within SLA	0	44	20180403
17	E137708	2018-12-23	14:03:00	Tech Support	1069	0.95	5	Within SLA	0	40	20181223
18	M855788	2018-04-17	08:00:00	Tech Support	758	0.95	0	Within SLA	0	24	20180417
19	E778362	2018-11-02	08:39:00	Billing	838	1.20	0	Within SLA	0	62	20181102
20	N119221	2018-11-22	14:44:00	Tech Support	681	0.95	10	Within SLA	0	23	20181122
21	V228277	2018-04-08	11:51:00	Billing	1272	1.20	42	Outside SLA	0	52	20180408
22	T398672	2018-11-07	14:34:00	Tech Support	252	0.95	0	Within SLA	0	47	20181107
23	Q921541	2018-11-24	11:48:00	Tech Support	84	0.95	271	Outside SLA	0	27	20181124
24	E243130	2018-03-22	11:37:00	Billing	229	1.20	132	Outside SLA	0	9	20180322
25	U641256	2018-01-25	17:00:00	Sales	432	1.52	6	Within SLA	0	36	20180125
26	M400222	2018-11-21	16:20:00	Billing	821	1.20	23	Within SLA	0	37	20181121

Query executed successfully. MSI (15.0 RTM) | AzureAD\EvanKim (56) | Report\_CallCenter\_DWH | 00:00:00 | 98,975 rows

- The “EmployeeSurrKey” generated without any issue. The “DateKey” contain historical records.

Part 2. Searching missing values of “EmployeeID” and Aggregate group by Rejected values, then count their numbers. It is a important indicator to remark the duplicate data and the other unexpected issues.

- MLT - Employee Not Found :



- After “Multicast” SSIS component, two options will be applied one searching for the missing value through the two “Derived Columns” SSIS components, such as: “DER - Employee -1” and “DER - EmployeeRejects”. It facilitates automatization of SCD process.
- DER - Employee -1 : Derived Column Name : EmployeeSurrKey-1, Derived Column <add as new column> , Expression : -1

Derived Column Name	Derived Column	Expression	Data Type	Length
EmployeeSurrKey-1	<add as new column>	-1	four-byte signed integ...	

- This component will go to search the missing value
- **ALL - EmployeeSurrKey : “Union All” SSIS component receive the outcomes from “DER - Employee - 1”**

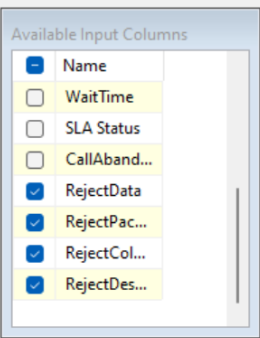
Output Column Name	Union All Input 1	Union All Input 2
EmployeeID	EmployeeID	EmployeeID
CallDate	CallDate	CallDate
CallTime	CallTime	CallTime
CallType	CallType	CallType
CallDuration	CallDuration	CallDuration
CallCharges	CallCharges	CallCharges
WaitTime	WaitTime	WaitTime
SLA Status	SLA Status	SLA Status
CallAbandoned	CallAbandoned	CallAbandoned
EmployeeSurrKey	EmployeeSurrKey	EmployeeSurrKey-1

### Part 3. Functional Rejects Table

- **DER - EmployeeRejects :**

Derived Column Name	Derived Column	Expression	Data Type	Length
RejectData	<add as new column>	GETDATE()	database timestamp [DT_DBTIMESTAMP]	
RejectPackageAndTask	<add as new column>	(DT_WSTR,100)@[System::PackageName] + "AND" + (DT_WSTR,100)@[System::TaskName]	Unicode string [DT_WSTR]	203
RejectColumn	<add as new column>	"EmployeeID"	Unicode string [DT_WSTR]	10
RejectDescription	<add as new column>	"The value" + (DT_WSTR,100)EmployeeID + "was not found in the dimension table"	Unicode string [DT_WSTR]	145

- **AGG - GroupBy :**



Input Column	Output Alias	Operation
RejectData	RejectData	Group by
RejectPackageAndTask	RejectPackageAndTask	Group by
RejectColumn	RejectColumn	Group by
RejectDescription	RejectDescription	Group by
(*)	Count all	Count all

- Aggregate the Rejects outputs with their counting number.

- OLE\_DST - FunctionalRejects :

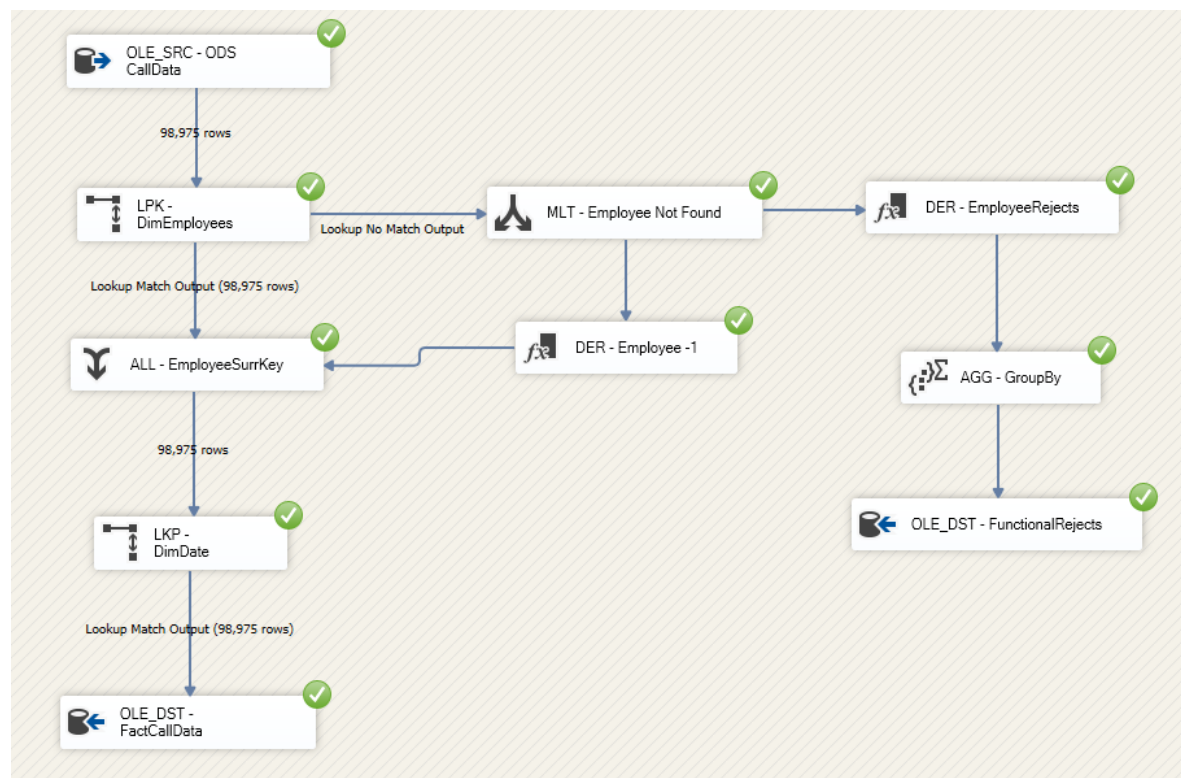
```

113 CREATE TABLE [FunctionalRejects] (
114     [RejectData] datetime,
115     [RejectPackageAndTask] nvarchar(203),
116     [RejectColumn] nvarchar(MAX),
117     [RejectDescription] nvarchar(MAX),
118     [Count all] numeric(20,0)

```

- As a precaution , set the maximum length to avoid the inconvenient cases for the data importing task.

- Result :



	EmployeeID	CallDate	CallTime	CallType	CallDuration	CallCharges	WaitTime	SLA Status	CallAbandoned	EmployeeSurrKey	DateKey
98960	X825147	2018-11-19	10:54:00	Tech Support	278	0.95	19	Within SLA	0	64	20181119
98961	X825147	2018-12-05	11:53:00	Billing	380	1.20	0	Within SLA	0	64	20181205
98962	X825147	2018-01-07	13:39:00	Tech Support	1366	0.95	29	Within SLA	0	64	20180107
98963	X825147	2018-12-06	14:29:00	Tech Support	917	0.95	11	Within SLA	0	64	20181206
98964	X825147	2018-02-20	17:56:00	Tech Support	528	0.95	0	Within SLA	1	64	20180220
98965	X825147	2018-04-27	19:16:00	Tech Support	1487	0.95	30	Within SLA	0	64	20180427
98966	X825147	2018-02-15	12:41:00	Tech Support	824	0.95	0	Within SLA	0	64	20180215
98967	X825147	2018-10-12	16:28:00	Tech Support	1378	0.95	0	Within SLA	0	64	20181012
98968	X825147	2018-09-14	13:28:00	Sales	433	1.52	0	Within SLA	0	64	20180914
98969	X825147	2018-12-08	12:24:00	Billing	108	1.20	0	Within SLA	0	64	20181208
98970	X825147	2018-10-29	13:18:00	Sales	802	1.52	0	Within SLA	1	64	20181029
98971	X825147	2018-03-26	19:22:00	Sales	361	1.52	0	Within SLA	0	64	20180326
98972	X825147	2018-02-11	16:13:00	Billing	884	1.20	23	Within SLA	0	64	20180211
98973	X825147	2018-09-21	15:30:00	Tech Support	183	0.95	217	Outside SLA	0	64	20180921
98974	X825147	2018-07-14	08:55:00	Tech Support	101	0.95	11	Within SLA	0	64	20180714
98975	X825147	2018-09-05	16:38:00	Tech Support	1447	0.95	18	Within SLA	0	64	20180905

Query executed successfully.

MSI (15.0 RTM) AzureAD\EvanKim (71) Report\_CallCenter\_DWH | 00:00:01 | 98,975 rows





# V. USE CASE

## 1. Manager Performance Evaluation

```
SELECT [e]. [ManagerName],
COUNT([fc].[EmployeeID]) AS [TotalCalls],
SUM(CASE WHEN [fc]. [CallAbandoned]='1' THEN 1 ELSE 0 END) AS [AbandonedCalls],
ROUND(AVG(CAST([fc]. [WaitTime] AS FLOAT)), 2) AS [AverageWaitTime],
ROUND(SUM(CAST([FC].[Call Charges]*[CallDuration]/60 AS FLOAT)), 2) AS [TotalCallCharges],
CAST((SUM(CASE WHEN [fc].[CallAbandoned] = '1' THEN 1 ELSE 0 END) *1.0/ COUNT([fc].[Call
Type])) AS DECIMAL(10, 3)) AS [AbandonRate]
FROM [dbo].[FactCallData] AS [fc]
INNER JOIN [dbo].[Employees] AS [e]
ON [fc]. [EmployeeID] = [e]. [EmployeeID]
GROUP BY [e]. [ManagerName]
ORDER BY [TotalCallCharges] DESC, [AbandonRate] ASC
```

### Objective:

Evaluate call center performance metrics by manager.

### Key Metrics:

**TotalCalls:** Total number of calls handled by each manager's team.

**AbandonedCalls:** Number of calls abandoned (where CallAbandoned = '1').

**AverageWaitTime:** Average wait time for calls, cast to float for accuracy.

**TotalCallCharges:** Total call charges, cast to float for accurate summation.

**AbandonRate:** Ratio of non-abandoned calls to total calls.

### Query Breakdown:

**SELECT:** Calculates and retrieves the performance metrics.

**FROM:** Uses DWH\_FactCallData (fc) and DWH\_Employees (e) tables.

**JOIN:** Combines call data with employee/manager information on EmployeeID.

**GROUP BY:** Aggregates results by ManagerName.

**ORDER BY :** TotalCallCharges in descending order, and AbandonRate in ascending order.

### Usage:

This query helps analyze the effectiveness of each manager's team in the call center by providing key performance metrics. It assists in identifying areas for improvement and making informed decisions

### Results:

	ManagerName	TotalCalls	AbandonedCalls	AverageWaitTime	TotalCallCharges	AbandonRate
1	Ardath Ducharme	10906	668	27.74	162621.07	0.061
2	Elsie Taplin	10872	643	27.21	162265.49	0.059
3	Jamar Prael	10893	634	29.15	162106.45	0.058
4	Casey Bainbridge	10784	625	26.11	160049.12	0.058
5	Collin Trotman	9444	598	27.13	138571.15	0.063
6	Rana Taub	9242	530	27.6	135679.09	0.057
7	Gonzalo Lesage	7701	492	28.24	113796.97	0.064
8	Shala Lion	7667	425	27.08	113594.3	0.055
9	Nova Harshberger	6211	383	26.74	93290.83	0.062
10	Abbie Leadbetter	6171	369	27.16	90563.16	0.060
11	Miyoko Degraw	4611	288	26.76	70329.09	0.062
12	Deidre Robbs	4473	285	24.92	65474.91	0.064

We can see that Ardath Ducharme is the manager that has the highest call charges (162K across three years), which Deidre Robbs is the one who performs worst.

## 2. Site Performance Evaluation

```

SELECT
    [e].[Region],
    [e].[Site],
    COUNT(*) AS [TotalCalls],
    AVG([f].[CallDuration]) AS [AvgCallDuration],
    SUM([f].[CallCharges]*[CallDuration]/60) AS [TotalCallCharges],
    SUM(CASE WHEN [f].[CallAbandoned] = '1' THEN 1 ELSE 0 END) AS [nb_abandon],
    CAST(SUM(CASE WHEN [f].[CallAbandoned] = '1' THEN 1 ELSE 0 END) / CAST(COUNT(*) AS
decimal(10,4)) AS decimal(10,4)) AS [AbandonRate],
    SUM(CASE WHEN [f].[SLA Status] = 'Within SLA' THEN 1 ELSE 0 END) AS [nb_within_sla],
    CAST(SUM(CASE WHEN [f].[SLA Status] = 'Within SLA' THEN 1 ELSE 0 END) /
CAST(COUNT(*) AS decimal(10,4)) AS decimal(10,4)) AS SLARate
FROM
    [dbo].[FactCallData] [f]
JOIN
    [dbo].[Employees] [e] ON [f].[EmployeeID] = [e].[EmployeeID]
GROUP BY
    [e].[Region], [e].[Site]
ORDER BY
    [TotalCalls] DESC;

```

### Objective:

Evaluate call center performance metrics by region and site.

### Key Metrics:

**TotalCalls:** Total number of calls handled in each region and site.

**AvgCallDuration:** Average duration of calls, providing insight into the handling efficiency.

**TotalCallCharges:** Total charges for calls, indicating the cost associated with the calls.

**Nb\_abandon:** the number of abandon calls.

**AbandonRate:** Ratio of abandoned calls to total calls, highlighting the rate at which calls are not completed.

**Nb\_within\_SLA:** the number of calls within SLA.

**SLARate:** Ratio of calls handled within the Service Level Agreement (SLA) to total calls, indicating compliance with service standards.

**Query Breakdown:**

**SELECT:** Calculates and retrieves the performance metrics for each region and site.

**FROM:** Uses DWH\_FactCallData (aliased as f) and DWH\_Employees (aliased as e) tables.

**JOIN:** Combines call data with employee information on EmployeeID.

**GROUP BY:** Aggregates results by Region and Site.

**ORDER BY:** Sorts the results by the total number of calls in descending order to highlight the busiest regions and sites.

**Usage:**

This query helps analyze the performance of each region and site in the call center by providing key performance metrics. It assists in identifying high-performing areas and regions or sites that may need additional support or resources, thereby enabling informed decision-making to improve overall efficiency and service quality.

**Results:**

	Region	Site	TotalCalls	AvgCallDuration	TotalCallCharges	nb_abandon	AbandonRate	nb_within_sla	SLARate
1	South	Jacksonville, FL	35562	746	527030.315170	2154	0.0606	31366	0.8820
2	West	Aurora, CO	33992	749	504829.714187	1981	0.0583	30022	0.8832
3	West	Spokane, WA	29421	747	436481.598468	1805	0.0614	25909	0.8806

South region has the highest number of total calls and call charges.

### 3. Year-on-year Evaluation

```

WITH CTE AS (
    SELECT YEAR([CallDate]) AS year
           , COUNT(*) AS nb_calls
           , CAST(SUM([CallDuration]*[CallCharges]/60) AS INT) AS total_charge_dollar
    FROM [CallCenter_DWH].[dbo].[FactCallData] [f]
    GROUP BY YEAR([CallDate])
)

SELECT year
       , nb_calls
       , LAG(nb_calls) OVER (ORDER BY year) AS nb_calls_last_year
       , nb_calls - LAG(nb_calls) OVER (ORDER BY year) AS nb_calls_yoy_diff
       , total_charge_dollar
       , LAG(total_charge_dollar) OVER (ORDER BY year) AS total_charge_dollar_last_year
       , total_charge_dollar - LAG(total_charge_dollar) OVER (ORDER BY year) AS
total_charge_dollar_yoy_diff
FROM CTE

```

## Objective:

To evaluate performance of each year and detect if there's positive yearly growth.

## Methodology:

A CTE (common table expression) is used and serves as an intermediate table to get the number of calls and total charges made for each year. Here a formular  $[\text{CallDuration}] * [\text{CallCharges}] / 60$  is designed to calculate the charges made by each call.

Later in the output two new columns (last\_year and yoy\_diff) show up to compare and calculate between this year's data and last year's for both nb\_calls and total\_charge.

## Result:

	year	nb_calls	nb_calls_last_year	nb_calls_yoy_diff	total_charge_dollar	total_charge_dollar_last_year	total_charge_dollar_yoy_diff
1	2018	33057	NULL	NULL	463264	NULL	NULL
2	2019	32987	33057	-70	488655	463264	25391
3	2020	32931	32987	-56	516421	488655	27766

We can find that although the number of calls is declining year by year, but the total charge is increasing, which indicates the company is performing well.

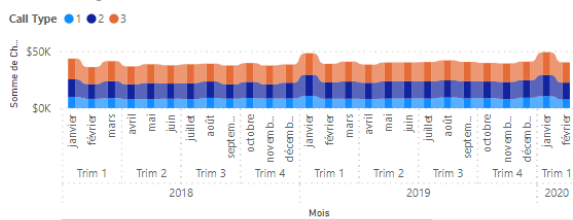
## 4. Power BI Dashboard

### Data Centre Call Services Dashboard

Sum of Charges by Managers

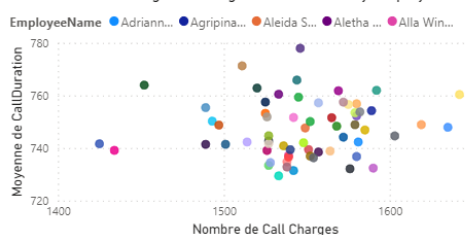


Sum of Charges Evolution



Region: Tout 01/01/2018 31/12/2020

Number of Call Charges and Avg de CallDuration by Employees



Average of Call Duration by Site



After connecting the SQL server with Power BI dashboard we could make simple and real-time data analysis on the dataset. In simple conclusions, we can observe that January tends to be the busiest season in which the call centre gets the most profit compared to

other months. Ardath is the manager who has the best performance in terms of the call charges surpassing others; and Ardath is also the employee who has the second the greatest number of calls. Other information and use cases needs could be fulfilled as well by the real-time monitoring in Power BI.

## VI. CONCLUSION

We have made some choices during the ETL process:

### FactCallData

This is our final fact table. We set it to be fact table because:

- It has quantitative data such as call duration, wait time and call charge, and contextual attributes such as call type, SLA status and call abandoned.
- It has transactional grain: each row in the table represents a single call, providing detailed, transaction-level data.
- It stores foreign keys to dimensional tables: EmployeeID and EmployeeSurrKey as foreign keys to dimensional table Employees.
- CallChargeID enables the calculation of call charges with this table.
- CallTypeLable facilitates the categorization of each call records so that we could know to which type it belongs such as "Sales", "Biling", "Tech Support", making it more straightforward for users.

It starts from table ODS\_CallData in the ODS step where we lookuped the tables CallType and CallCharges from STA and took the fields we think useful (call type and call charges). We combined it with our raw CallData table from the STA step. Nevertheless, as requested we also added the field SLA Status, which specify is if a call is within or outside SLA, in the ODS step and finally in DWS.

Therefore, this fact tale FactCallData that has every field concerning the calling data and considering its foreign keys, it can thus support various analyses, such as performance analysis, cost analysis and trend analysis, as what we have shown in the user cases.

### Employees

It's a dimensional table. The role of dimensional table is mainly to provide context to the quantitative data store in fact table, and we can find that our table Employees perfectly fits this role. It starts from the Employees table in STA, and we lookuped the geographical table, US States, to reduce complexity.

By embedding attributes such as StateName, Region, and Site directly within the DWH\_DimEmployees table, we ensure that all relevant geographic information is readily

available alongside employee data. This integration facilitates quicker access to comprehensive datasets, enabling more efficient querying and analysis.

This approach not only simplifies our data model but also enhances the performance of our queries, reducing the complexity and potential for errors in our analyses.

Consequently, analysts can now perform site-specific performance evaluations with greater ease and accuracy, supporting more informed decision-making across various organizational levels.

### **Future expansion**

We believe that our project is a practical case of building an entire ETL process from scratch. If we have call data for future years (year 2021, 2022 etc.), we can just easily add them into the folder in the STA Call Data step, and the whole pipeline will still be running well.